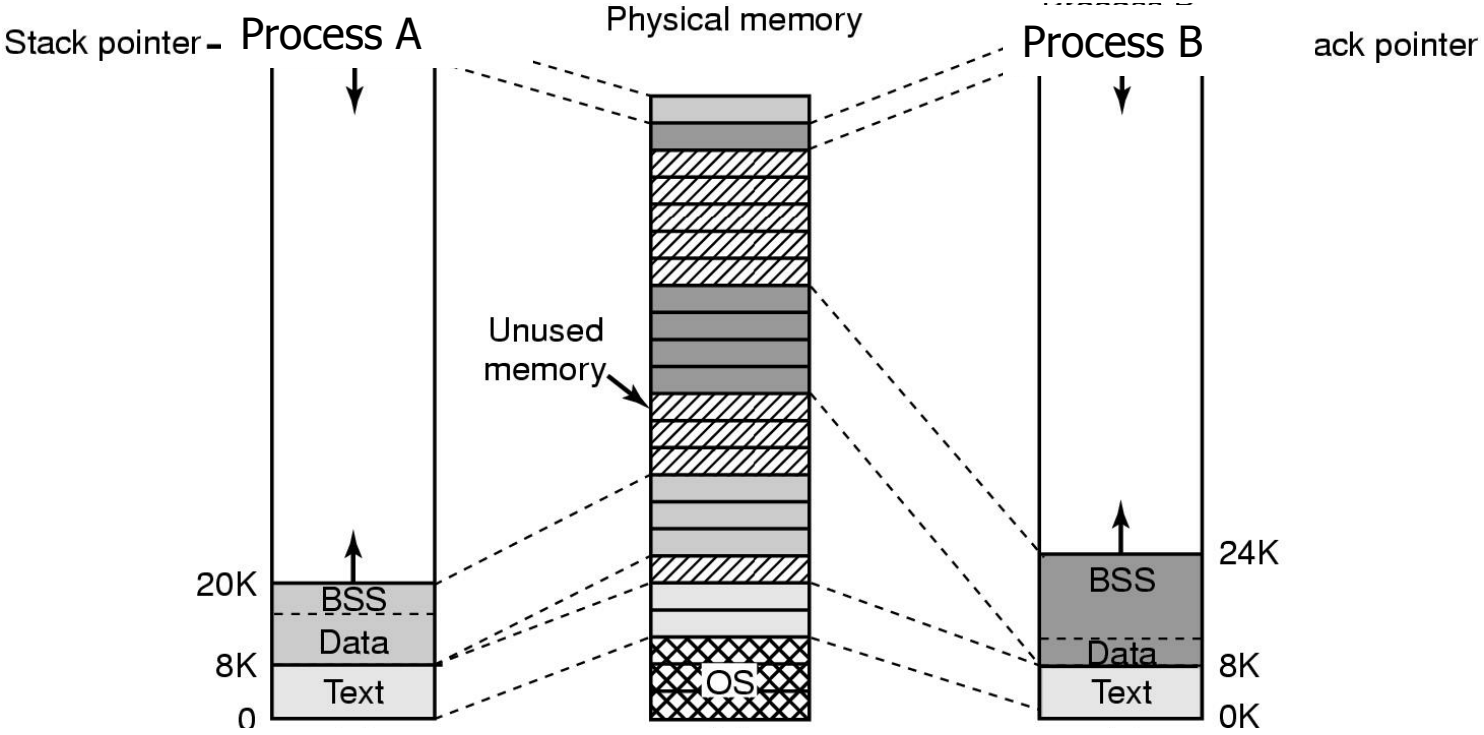
Capitolo 4

Gestione della Memoria

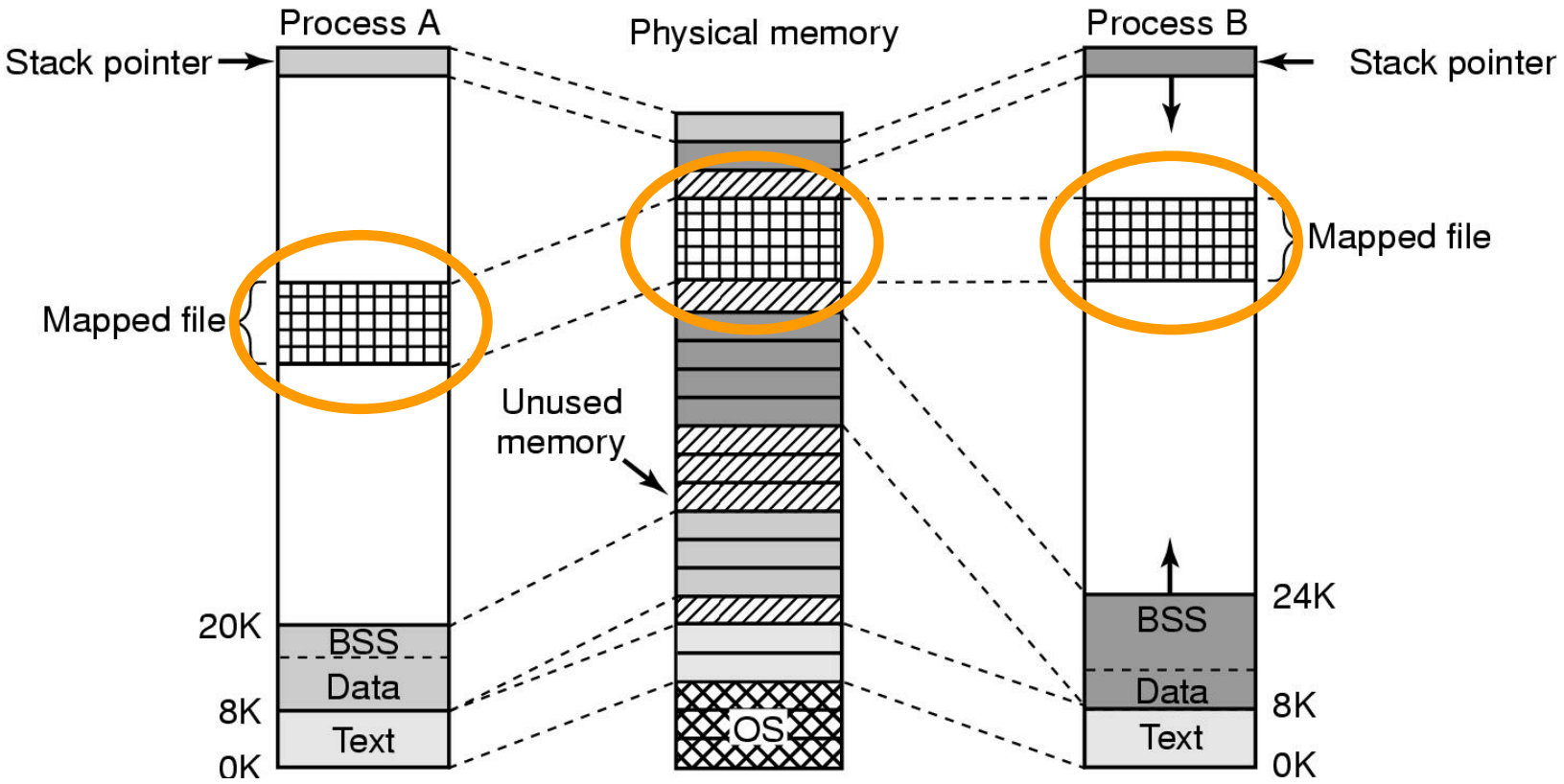
4.1 Introduzione alla gestione della memoria
4.2 Swapping
4.3 Memoria virtuale
4.4 Implementazione
4.5 Algoritmi di sostituzione
4.6 Criteri di progetto per la paginazione
4.7 Case study: Unix
4.8 Case study: Windows 2000

Organizzazione della Memoria



- Spazio logico del processo A
- Memoria fisica
- Spazio logico del processo B

Condivisione di Files



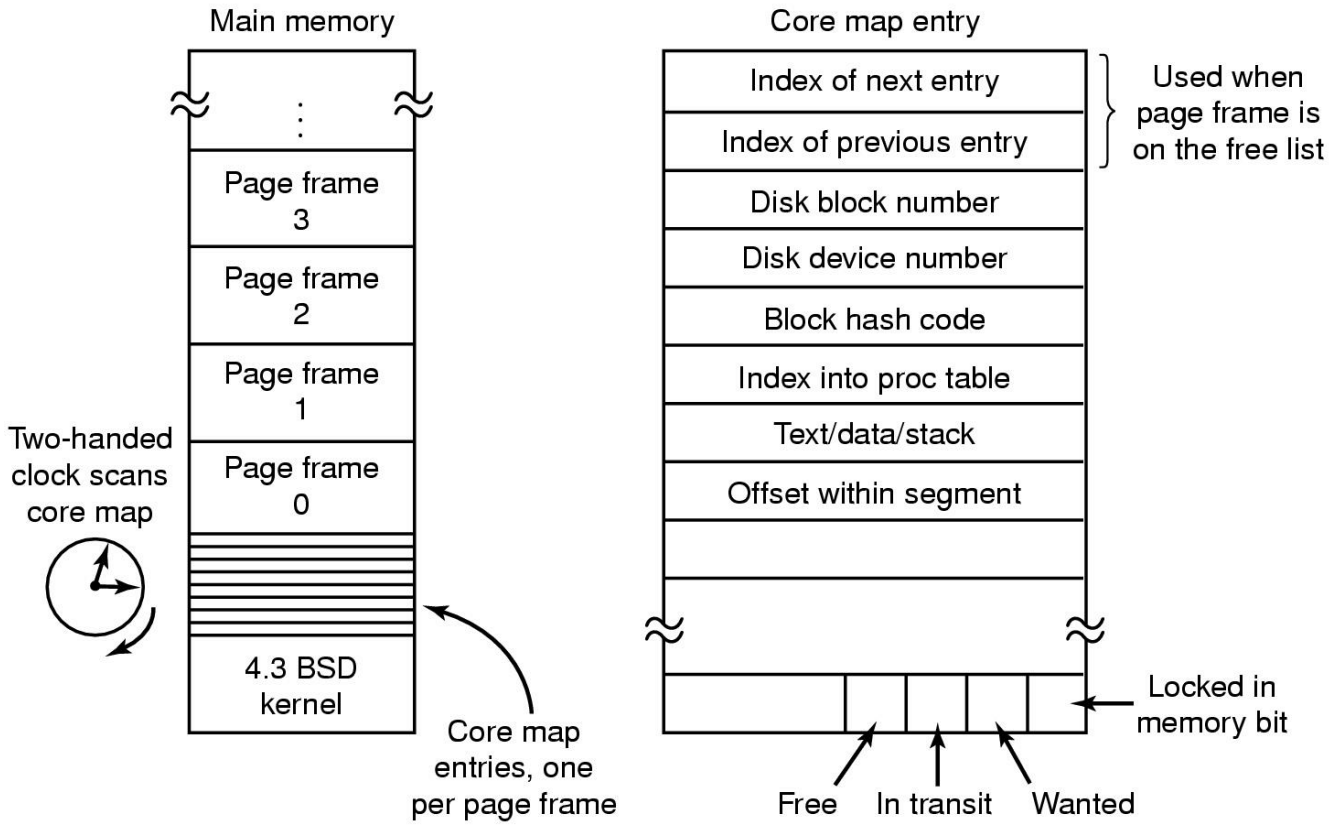
Un file mappato simultaneamente in due processi

System Calls per la Gestione della Memoria

System call	Description
s = brk(addr)	Change data segment size
a = mmap(addr, len, prot, flags, fd, offset)	Map a file in
s = unmap(addr, len)	Unmap a file

- **s** è un codice di errore
- **b** e **addr** sono indirizzi di memoria
- **len** è una lunghezza
- **prot** controlla la protezione
- **flags** bit vari
- **fd** è un descrittore di file
- **offset** è un offset su un file

Paginazione in UNIX

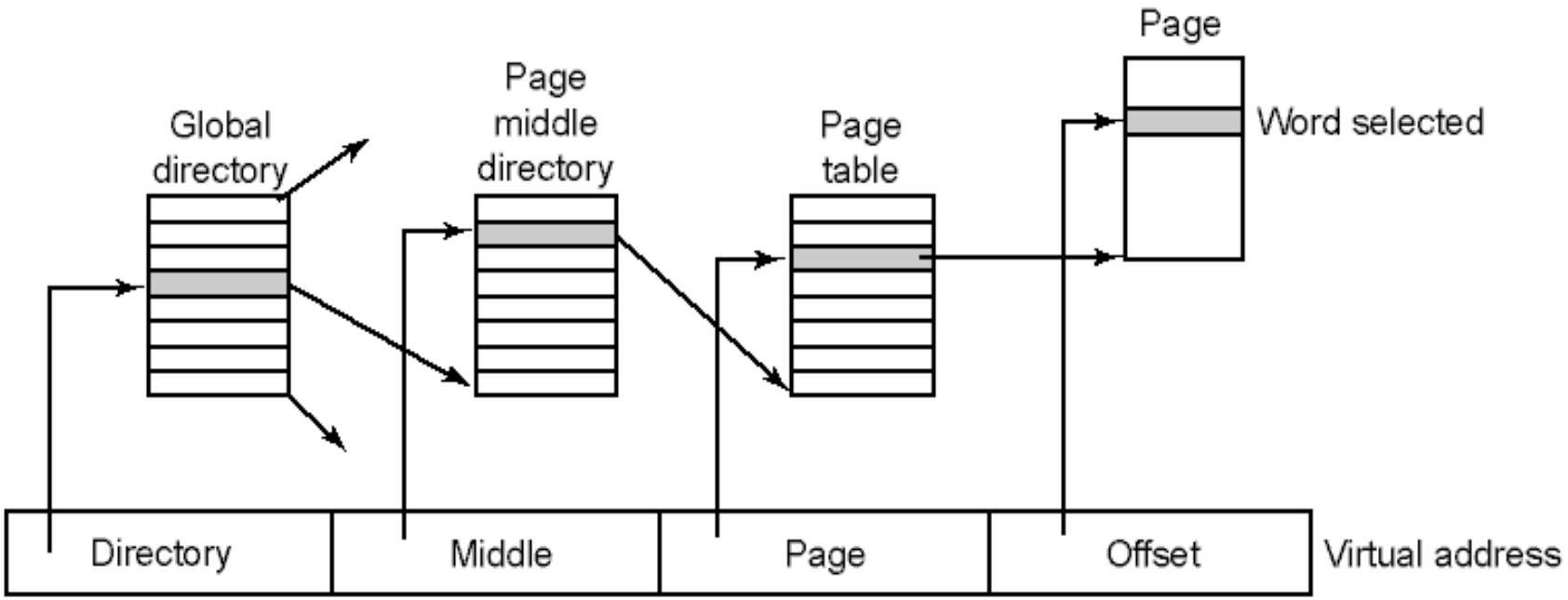


La tabella principale ha una entry per ogni pagina

Algoritmo di Sostituzione in UNIX (BSD)

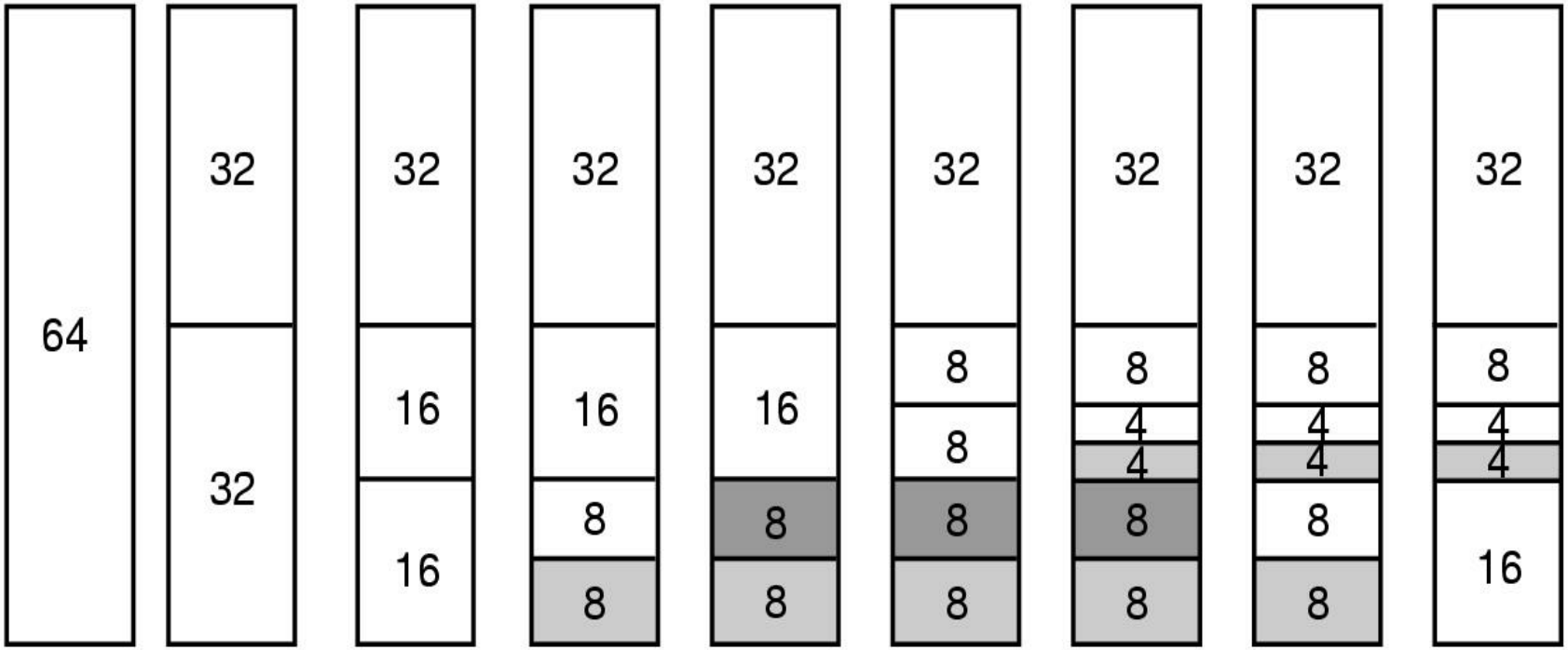
- Eseguito dal page daemon
  - Cerca di mantenere almeno *lostfree* pagine libere
  - Usa l'algoritmo Two-handed clock
    - una variante dell'algoritmo Clock
    - Con allocazione globale
- Se non è possibile mantenere *lostfree* pagine libere si fa lo swapping di qualche processo
  - Si fa lo swapping di processi idle da più tempo e in alternativa di processi grossi
  - Swap-in se ci sono pagine libere a sufficienza
    - Il processo viene scelto in base a vari parametri

Paginazione in Linux (1)



Linux usa tabelle delle pagine a tre livelli

Paginazione in Linux (2)



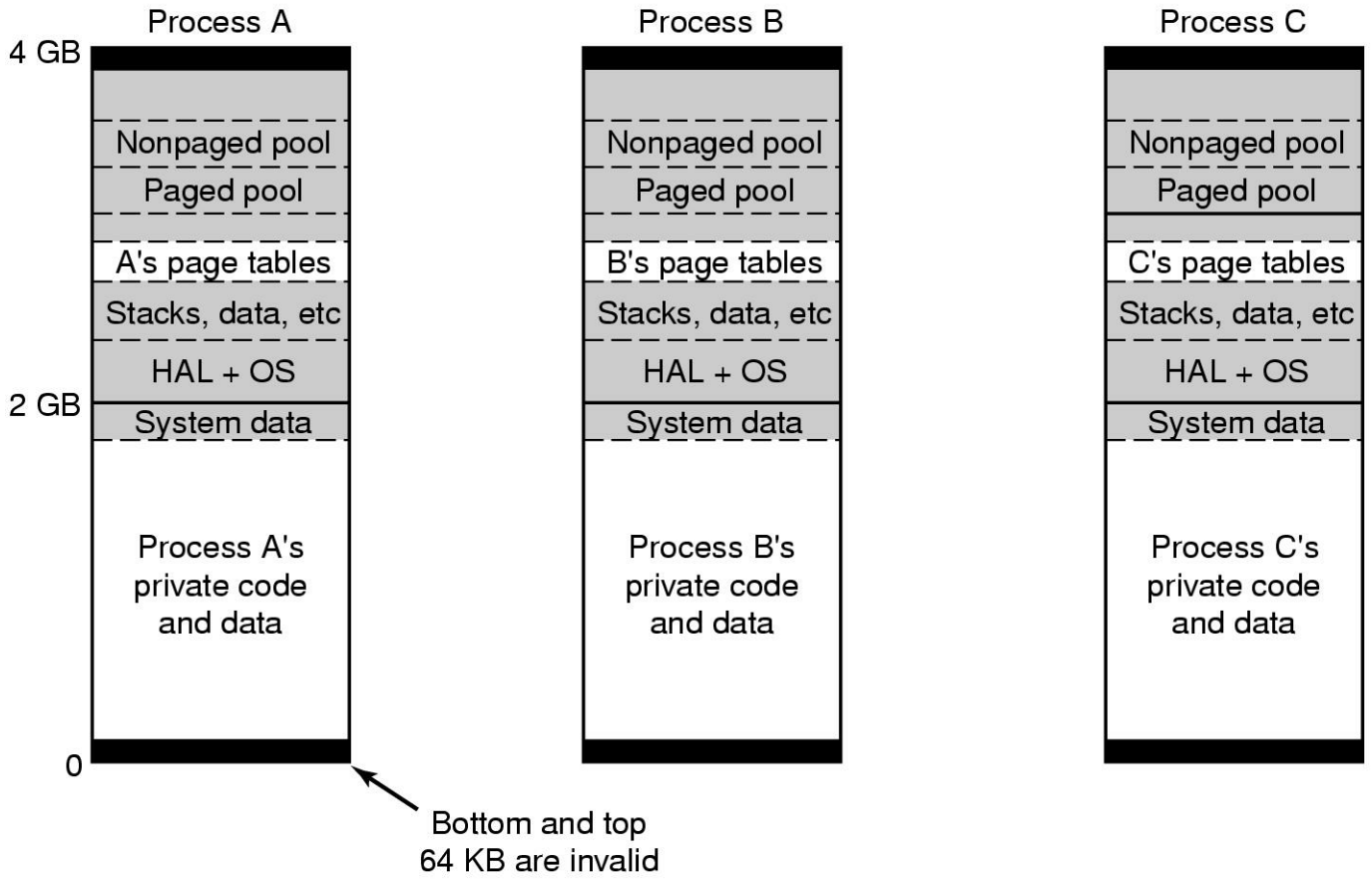
Algoritmo Buddy

Capitolo 4

Gestione della Memoria

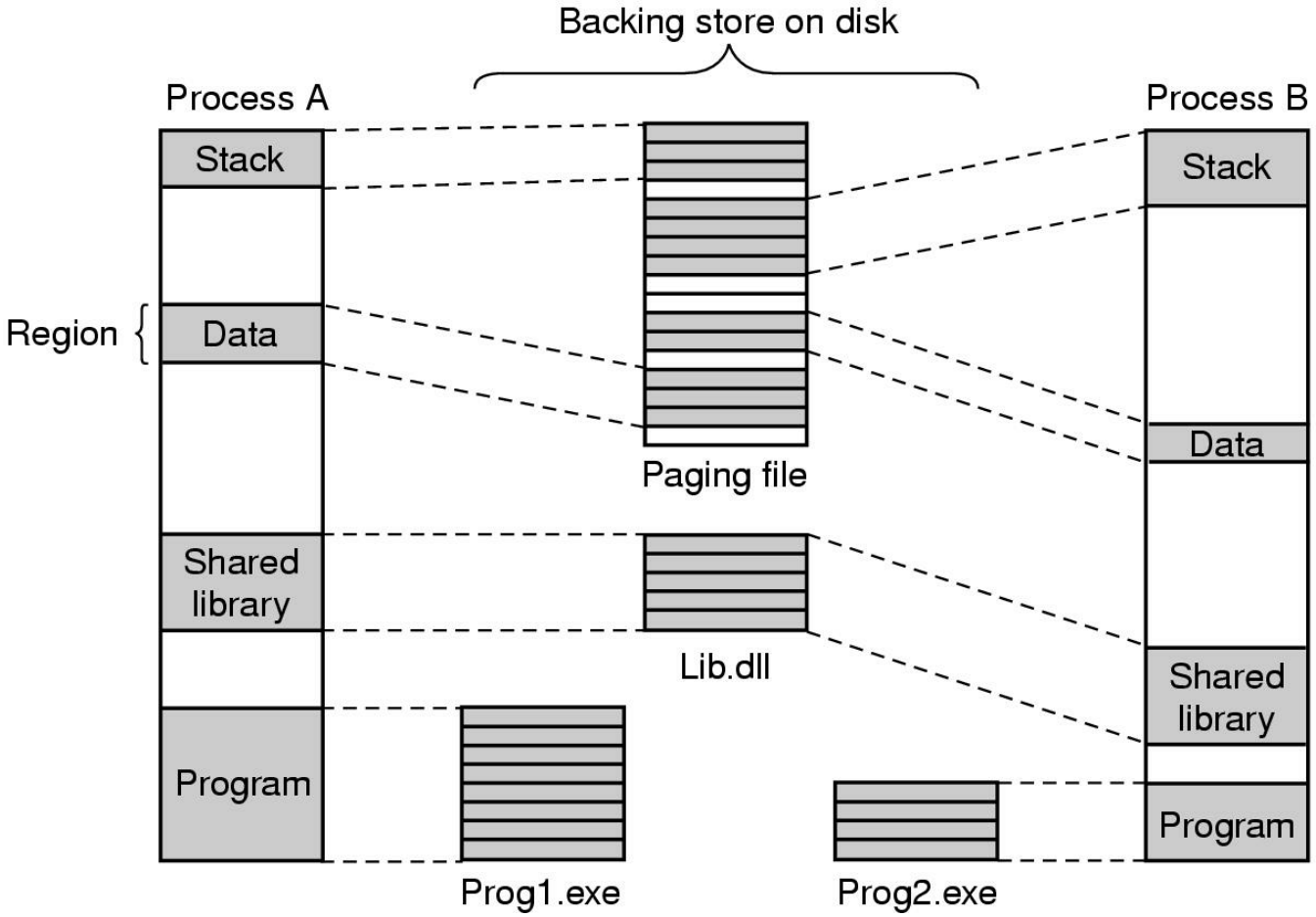
4.1 Introduzione alla gestione della memoria
4.2 Swapping
4.3 Memoria virtuale
4.4 Implementazione
4.5 Algoritmi di sostituzione
4.6 Criteri di progetto per la paginazione
4.7 Case study: Unix
4.8 Case study: Windows 2000

Concetti Fondamentali (1)



- Layout di spazi di indirizzamento logici per 3 processi
- Le aree bianche sono private
- Le aree scure sono condivise

Concetti Fondamentali(2)



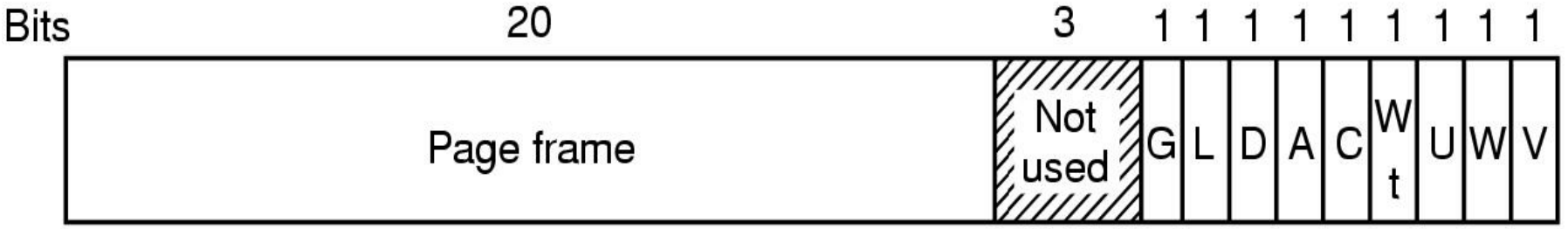
- Aree mappate con le rispettive pagine su disco
- Il file *lib.dll* è mappato in due spazi di indirizzamento contemporaneamente

System Calls per la Gestione della Memoria

Win32 API function	Description
VirtualAlloc	Reserve or commit a region
VirtualFree	Release or decommit a region
VirtualProtect	Change the read/write/execute protection on a region
VirtualQuery	Inquire about the status of a region
VirtualLock	Make a region memory resident (i.e., disable paging for it)
VirtualUnlock	Make a region pageable in the usual way
CreateFileMapping	Create a file mapping object and (optionally) assign it a name
MapViewOfFile	Map (part of) a file into the address space
UnmapViewOfFile	Remove a mapped file from the address space
OpenFileMapping	Open a previously created file mapping object

Le principali funzioni delle API Win32 per la gestione della memoria in Windows 2000

Implementazione della Gestione della Memoria



G: Page is global to all processes
L: Large (4-MB) page
D: Page is dirty
A: Page has been accessed

Wt: Write through (no caching)
U: Page is accessible in user mode
W: Writing to the page permitted
V: Valid page table entry

Una entry della tabella delle pagine sul Pentium

Algoritmo di Sostituzione in Windows

- Basato sull'algoritmo working set
  - Con parametri min e max dimensione del working set
  - Allocazione (fondamentalmente) locale
- Se la dimensione del working set corrente è *x*, in caso di page fault:
  - Se x<min una pagina è caricata
  - Se x>max si riduce la dimensione dell'working set
- Utilizza anche il demone "balance set manager" per mantenere un certo numero di pagine libere